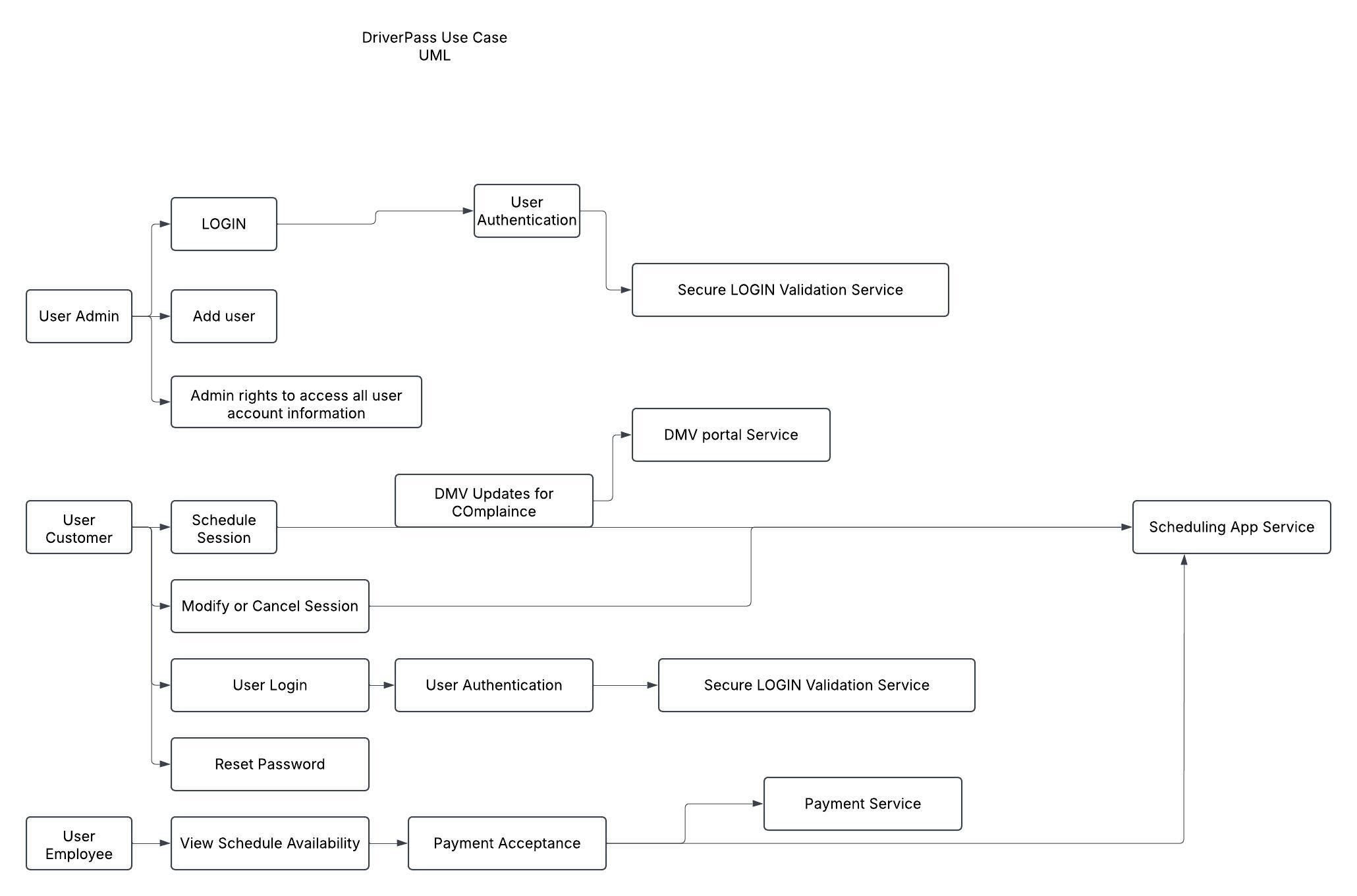
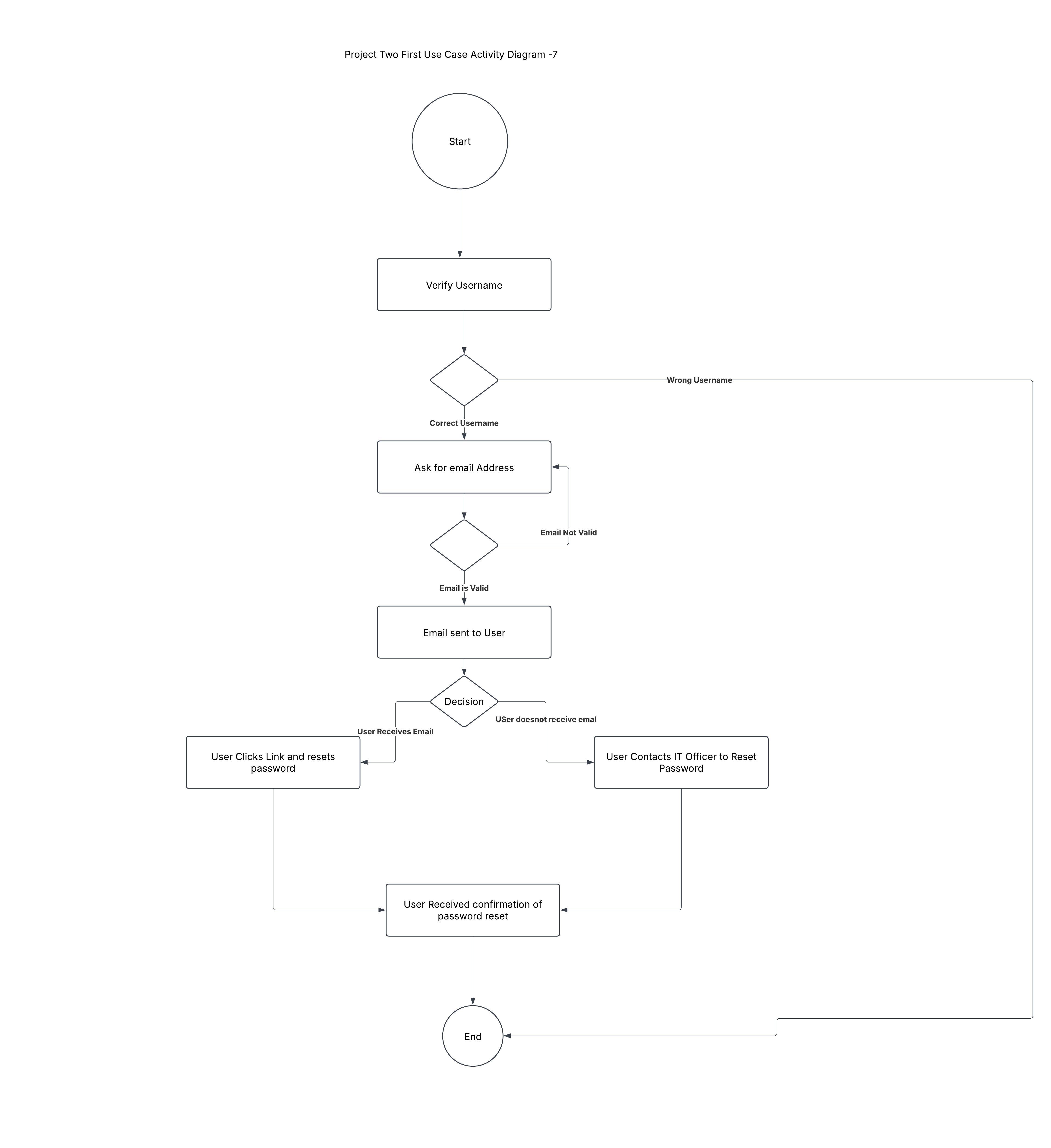
# CS 255 System Design Document

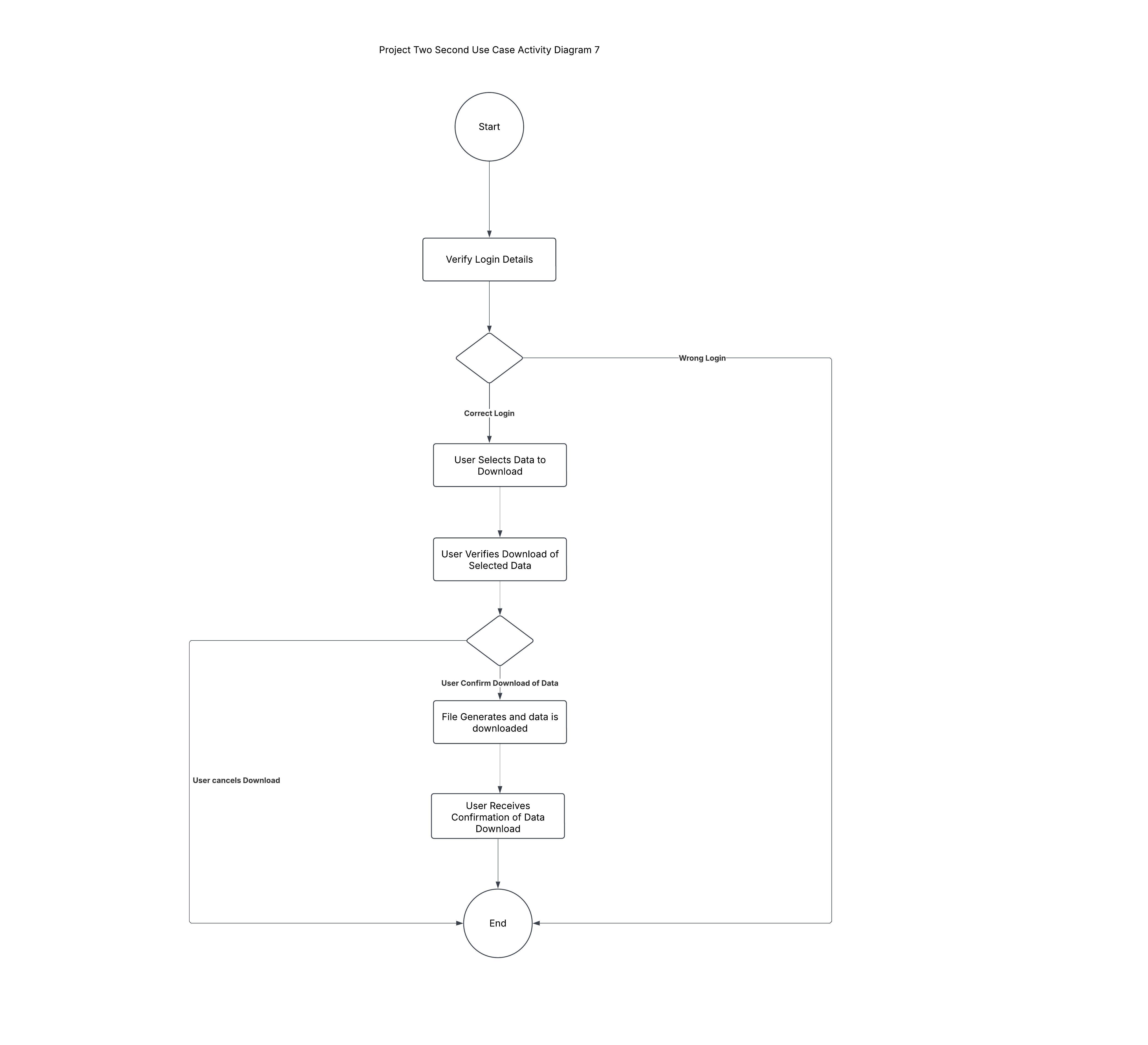
## UML Diagrams

### UML Use Case Diagram

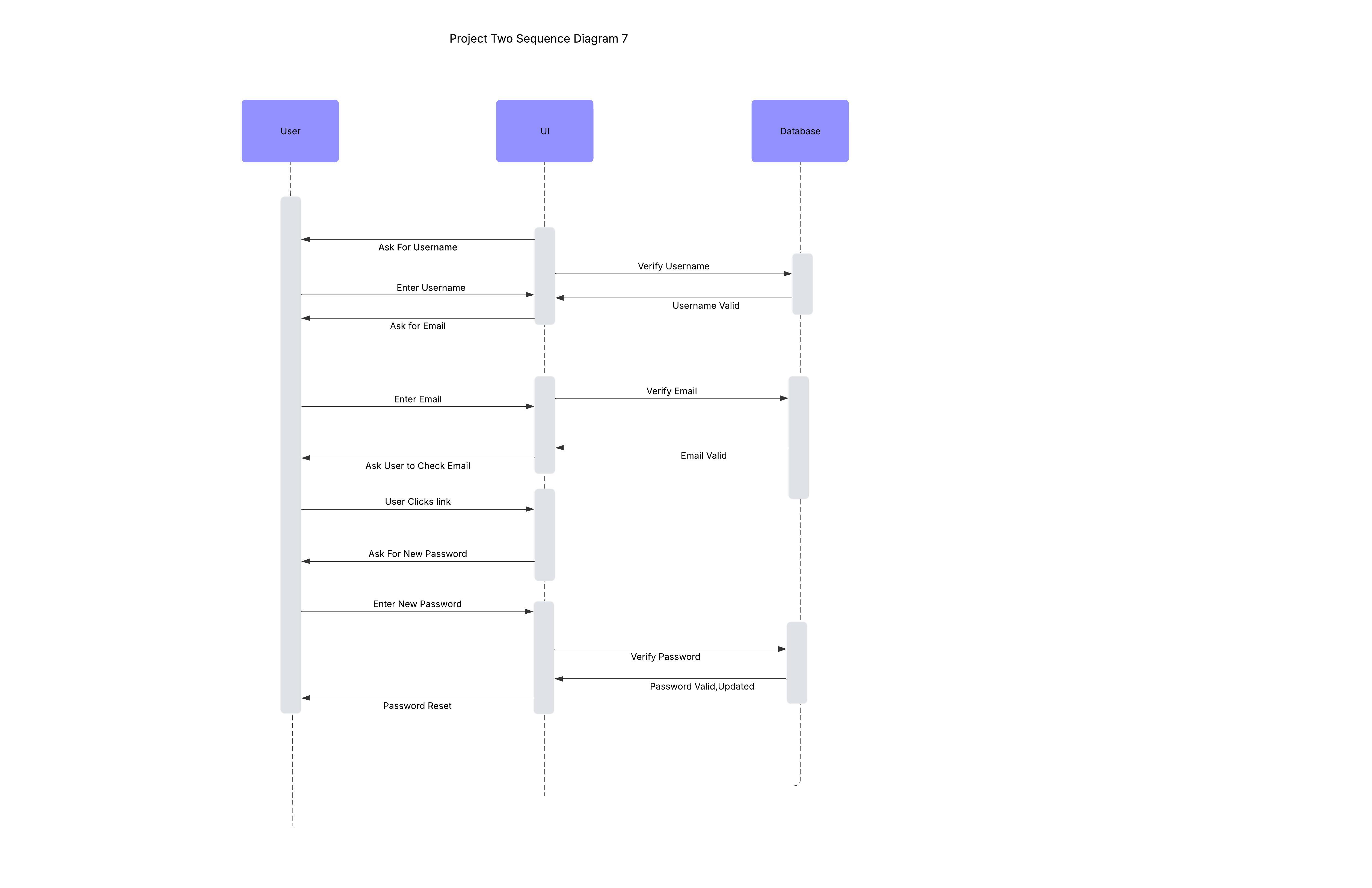


### UML Activity Diagrams

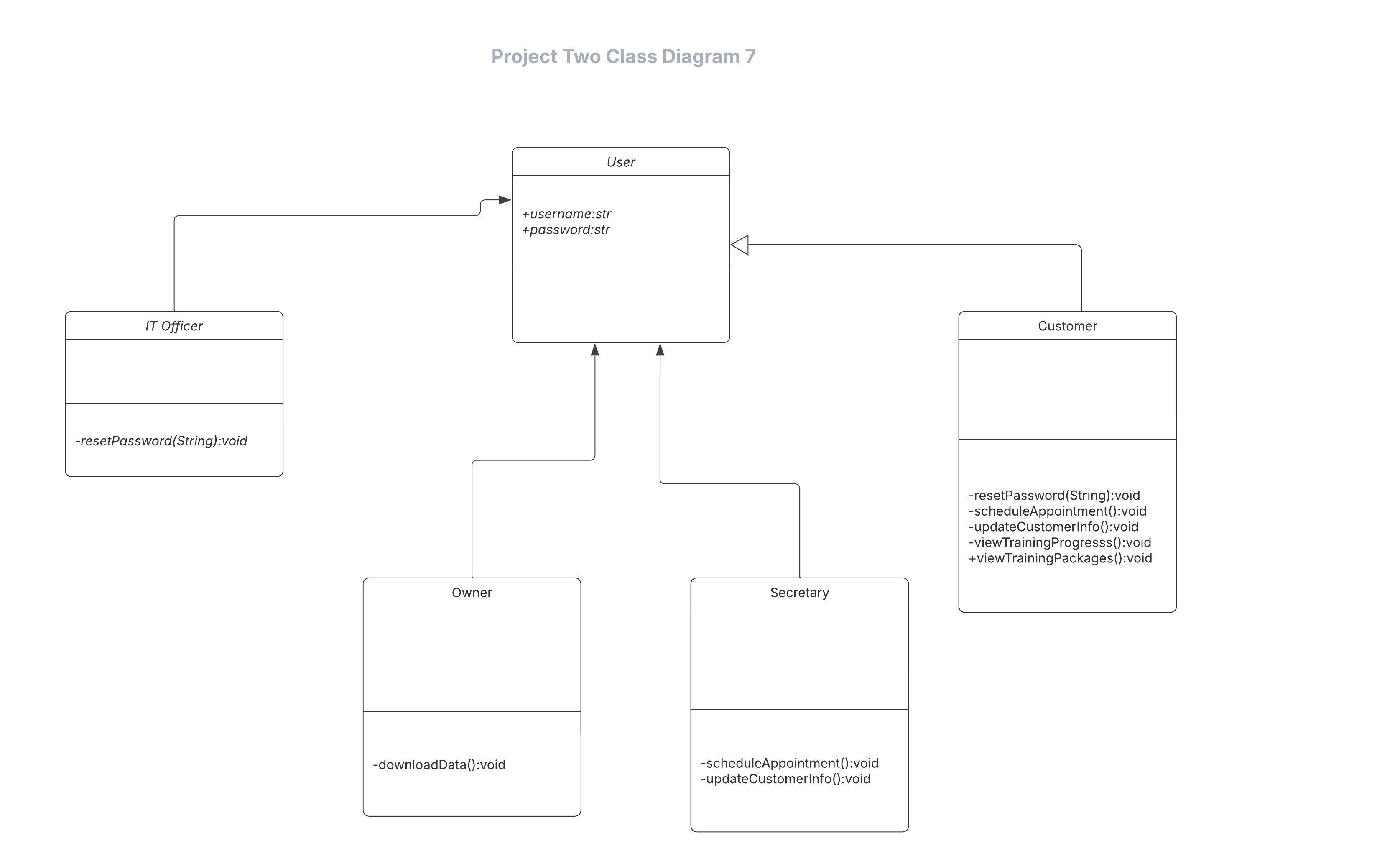




### UML Sequence Diagram



### UML Class Diagram



## Technical Requirements

DriverPass will demand a robust and scalable technical base for scheduling and managing driving lessons. It will be cloud-based, on companies like AWS, Azure, or Google Cloud, ensuring reliability and scalability for remote access. It will employ load balancers to manage traffic effectively and sustain performance during peak demands. Further, data will be backed up with the highly secure basic cloud data storage to allow for rapid recovery in the event of a system failure. The system is expected to interface with various devices inclusive of PCs, laptops, tablets, and mobile phones across differing operating systems.

The system will utilize React.js for a responsive and friendly user interface at the frontend, while the backend includes Node.js that manages all business logic and server-side execution. The system would apply a relational database like PostgreSQL or MySQL for secure user information, class schedule, and payment transaction records. Payment processing shall be done through secure payment gateways such as PayPal or Stripe and will be compliant with the necessary standards. The system will employ either OAuth2.0 or JWT authentication to ensure secure access to the users. Also, RESTful APIs will enhance communication amongst system components and allow real-time updates, including reception of changes of DMV policies.

DriverPass will enforce security and should implement multiple strategies to protect user data. The roles of the users will be defined sufficiently enough that they will find it too hard or unsanctioned to perform a task for which they haven't been authorized under role-based access control RBAC. Sensitive information, including passwords and payments, will be protected by AES-256 encryption. Also, password authentication will be confined using bcrypt or Argon2 hashing techniques. The whole system will descend into SSL/TLS encryption to maintain the safety of data in transfers. Lastly, an audit logging system will be in place that tracks users and changes made to the class schedules. This provides a clear overview for accountability.

We shall proceed to create the infrastructures for its scalable support dictated by its sustainability. The system shall be entirely hosted on AWS EC2 instances or Google Cloud Compute Engine deploying Kubernetes or Docker containers to allow microservices architecture, which makes it easy for portions to scale on its own. Monitoring and logging systems such as CloudWatch, Loggly, or Splunk will be incorporated to track performance, diagnose anomalies and send alerts. Furthermore, the implementation of disaster recovery, automated daily backups will continue to restore functionality with minimal downtime in the event of unforeseen failures.

These technical requirements guarantee that the DriverPass system will remain efficient, secure, and easily scalable to sufficiently meet the client's needs for a ground-up user-friendly, powerful driving lesson administration platform.